

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	25	(application program\$4 software) with (lifecycle life-cycle (life adj cycle)) with (run\$4 execut\$) with state	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:44
L2	103	((application program\$4 software) with (cycle lifecycle life-cycle (life adj cycle)) with (run\$4 execut\$) with state) same (stat\$3 with (activ\$5 paus\$3 load\$3 destroy\$3))) and ((program\$4 software application) with (manag\$5 control\$4) with (run\$4 execut\$7))	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:43
L3	143	((application program\$4 software) with (cycle lifecycle life-cycle (life adj cycle)) with (run\$4 execut\$) with state) same (stat\$3 with (activ\$5 paus\$3 load\$3 destroy\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:42
L5	5	((application program\$4 software) with (lifecycle life-cycle (life adj cycle)) with (run\$4 execut\$) with state) same ((manag\$5 control\$4) with (activ\$5 paus\$3 load\$3 destroy\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:47
L6	14	((application program\$4 software) with (lifecycle life-cycle (life adj cycle)) with (run\$4 execut\$) with state) with ((manag\$5 control\$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:57
L7	2	l3 and 718/100.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:58
L8	1	l6 and 718/100.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:58
L9	0	l6 and 718/104.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:58
L10	2	l3 and 718/104.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:58
L11	1	l3 and 717/166.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:59

EAST Search History

L12	2	l6 and 717/166.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT ; IBM_TDB	OR	ON	2006/11/22 10:59
-----	---	----------------------	--	----	----	------------------



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction](#)

Terms used

application or **program** or **software** **paragraph** **lifecycle** **near/5** **run** or **execute** **near/4** **state** **near/4** **manage** or

Sort results by

[Save results to a Binder](#)

Try an [Advanced Search](#)

Display results

[Search Tips](#)

Try this search in [The ACM C](#)

☐ Open results in a new window

Results 101 - 120 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance

101 [Migrating well engineered Ada 83 applications into newer architecture and reuse based Ac](#)

[systems: experiences from Boeing's reuse initiative project](#)

Scott Arthur Moody

December 1996 **Proceedings of the conference on TRI-Ada '96: disciplined software develo
with Ada**

Publisher: ACM Press

Full text available: [pdf\(1.25 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

102 [Modeling, evaluation, and testing of paradyn instrumentation system](#)

Abdul Waheed, Diane T. Rover, Jeffrey K. Hollingsworth

November 1996 **Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROI
Supercomputing '96**

Publisher: IEEE Computer Society

Full text available: [pdf\(225.73 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index ter](#)

This paper presents a case study of modeling, evaluating, and testing the data collection service (an instrumentation system) of the Paradyn parallel performance measurement tool using well-k performance evaluation and experiment design techniques. The overall objective of the study is modeling- and simulation-based evaluation to provide feedback to the tool developers to help tl system configurations and task scheduling policies that can significantly reduce the ...

103 [XXL: a dual approach for building user interfaces](#)

Eric Lecolinet

November 1996 **Proceedings of the 9th annual ACM symposium on User interface software
technology**

Publisher: ACM Press

Full text available: [pdf\(1.96 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: distributed interfaces, interface builders, iterative development, scripting language: and visual equivalence, user interface software

104 [Level II technical support in a distributed computing environment](#)



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction](#)

Terms used

application or **program** or **software** **paragraph** **lifecycle** **near/5** **run** or **execute** **near/4** **state** **near/4** **manage** or

Sort results by

Display results

[Save results to a Binder](#)

[Search Tips](#)

☐ Open results in a new window

Try an [Advanced Search](#)

Try this search in [The ACM C](#)

Results 81 - 100 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) **[5](#)** [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance

81 [Summary of the sigmetrics symposium on parallel and distributed processing](#)



Jeffrey K. Hillingsworth, Barton P. Miller

March 1999 **ACM SIGMETRICS Performance Evaluation Review**, Volume 26 Issue 4

Publisher: ACM Press

Full text available: [pdf\(1.17 MB\)](#)

Additional Information: [full citation](#), [index terms](#)

82 [A survey of structured and object-oriented software specification methods and techniques](#)



Roel Wieringa

December 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 4

Publisher: ACM Press

Full text available: [pdf\(605.26 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This article surveys techniques used in structured and object-oriented software specification methods and techniques are classified as techniques for the specification of external interaction and internal decomposition. The external specification techniques are further subdivided into techniques for specification of functions, behavior, and communication. After surveying the techniques, we survey the way they are used in structured and object-oriented methods and indicate ways in which they can be improved.

Keywords: languages

83 [Performance and dependability evaluation of scalable massively parallel computer system](#)



[conjoint simulation](#)

Axel Hein, Mario Dal Cin

October 1998 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 8 Issue 4

Publisher: ACM Press

Full text available: [pdf\(501.59 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Computer systems are becoming more and more a part of our daily life; business and industry rely on their service, and the health of human beings depends on their correct functioning. Computer systems used for critical tasks have to be carefully designed and tested during the early design stage, the prototype phase, and their operational life. Methods and tools are required to support and facilitate this vital task. In this article, we tackle the issue of system-level performance and dependability.

Keywords: fault-tolerant and large-scale computer systems, hierarchical model design, object-oriented

modeling, process-based simulation, timed Petri nets

84 Performance monitoring in a Myrinet-connected SHRIMP cluster



Cheng Liao, Margaret Martonosi, Douglas W. Clark

August 1998 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**

Publisher: ACM Press

Full text available: pdf(1.26 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

85 Guidance for the use of the Ada programming language in high integrity systems



B. A. Wichmann

July 1998 **ACM SIGAda Ada Letters**, Volume XVIII Issue 4

Publisher: ACM Press

Full text available: pdf(2.93 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper is the current result of a study by the ISO HRG Rapporteur group which is being circulated for comment. Many people have contributed to this, but those who have either attended two recent meetings of the group or have made substantial e-mail comments are: Praful V Bhansali (Boeing, USA), Alan J. Blythe (University of York, UK), Bernard Carre' (Praxis Critical Systems, UK), Dan Craigen (ORA, Canada), Johnson MoD, UK), Stephen Michell (Canada), Gilles Motet (DGEI/INSA, France), George Romanoski (Boeing, USA)

86 Version models for software configuration management



Reidar Conradi, Bernhard Westfechtel

June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2

Publisher: ACM Press

Full text available: pdf(483.54 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

After more than 20 years of research and practice in software configuration management (SCM), constructing consistent configurations of versioned software products still remains a challenge. This paper focuses on the version models underlying both commercial systems and research prototypes. It provides an overview and classification of different versioning paradigms and defines and relates fundamental concepts such as revisions, variants, configurations, and changes. In particular, we focus on the challenges of managing configurations of versioned software products.

Keywords: changes, configuration rules, configurations, revisions, variants, versions

87 Using metalevel techniques in a flexible toolkit for CSCW applications



Paul Dourish

June 1998 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(292.97 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Ideally, software toolkits for collaborative applications should provide generic, reusable components that are applicable in a wide range of circumstances, which software developers can assemble to produce applications. However, the nature of CSCW applications and the mechanics of group interaction present a significant problem. Group interactions are significantly constrained by the structure of the underlying infrastructure, below the level at which toolkits typically offer control. This article describes the challenges of designing such toolkits.

Keywords: consistency control, consistency guarantees, data distribution, divergency, metalevel programming, open implementation, software architecture

88 Workshop on compositional software architectures: workshop report

May 1998 **ACM SIGSOFT Software Engineering Notes**, Volume 23 Issue 3



Publisher: ACM Press

Full text available: [pdf\(2.91 MB\)](#) Additional Information: [full citation](#), [index terms](#)

89 The design, implementation, and evaluation of Jade



Martin C. Rinard, Monica S. Lam

May 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume :

Publisher: ACM Press

Full text available: [pdf\(576.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Jade is a portable, implicitly parallel language designed for exploiting task-level concurrency. Jade programmers start with a program written in a standard serial, imperative language, then use Jade constructs to declare how parts of the program access data. The Jade implementation uses this access information to automatically extract the concurrency and map the application onto the hardware. The resulting parallel execution preserves the semantics of the original serial program ...

Keywords: parallel computing, parallel programming languages

90 Self-adaptive software for signal processing



Janos Sztipanovits, Gabor Karsai, Ted Bapty

May 1998 **Communications of the ACM**, Volume 41 Issue 5

Publisher: ACM Press

Full text available: [pdf\(325.60 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

91 Architecture-based runtime software evolution

Peyman Oreizy, Nenad Medvidovic, Richard N. Taylor

April 1998 **Proceedings of the 20th international conference on Software engineering**

Publisher: IEEE Computer Society

Full text available: [pdf\(1.28 MB\)](#) [Publisher Site](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

92 Frameworks for component-based client/server computing



Scott M. Lewandowski

March 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: [pdf\(243.81 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

93 PELLPACK: a problem-solving environment for PDE-based applications on multicomputer



E. N. Houstis, J. R. Rice, S. Weerawarana, A. C. Catlin, P. Papachiou, K.-Y. Wang, M. Gaitatzes

March 1998 **ACM Transactions on Mathematical Software (TOMS)**, Volume 24 Issue 1

Publisher: ACM Press

Full text available: [pdf\(26.30 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The article presents the software architecture and implementation of the problem-solving environment (PSE) PELLPACK for modeling physical objects described by partial differential equations (PDEs). The scope of this PSE is broad, as PELLPACK incorporates many PDE solving systems, and some of these,

include several specific PDE solving methods. Its coverage for 1D, 2D, and 3D elliptic or parabolic problems is quite broad, and it handles some hyperbolic problems. Since a PSE should p ...

Keywords: PDE language, execution models, knowledge bases, libraries, parallel reuse method, problem-solving environments, programming-in-the-large, software bus

94 Controlling access in multiuser interfaces



Prasun Dewan, Honghai Shen

March 1998 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 5 Issue 1

Publisher: ACM Press

Full text available: pdf(182.07 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Traditionally, access control has been studied in the areas of operating systems and database management systems. With the advent of multiuser interfaces, there is a need to provide access to the user interface. We have developed a general framework for supporting access control in multiuser interfaces. It is based on the classical notion of an access matrix, a generalized editing-based model of user-application interaction, and a flexible model of user-user coupling. It has been de ...

Keywords: access control, collaboration, computer-supported cooperative work, groupware, privacy, security, structure editors, user interface management systems

95 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available: pdf(4.21 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, process diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial communication patterns ...

96 Ada-Java communication in ADEPT



Anthony Gargaro

November 1997 **Proceedings of the conference on TRI-Ada '97**

Publisher: ACM Press

Full text available: pdf(2.12 MB)

Additional Information: [full citation](#), [references](#), [index terms](#)

97 Assessing process-centered software engineering environments



Vincenzo Ambriola, Reidar Conradi, Alfonso Fuggetta

July 1997 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 6

Publisher: ACM Press

Full text available: pdf(342.52 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Process-centered software engineering environments (PSEEs) are the most recent generation of environments supporting software development activities. They exploit an representation of the process (called the process model that specifies how to carry out software development activities, the roles and tasks of software developers, and how to use and control software development tools. A process model is therefore a vehicle to better understand and communicate the process. If ...

Keywords: CASE, enabling technology, process modeling languages, process-centered software engineering environments, software process

98 Trace-driven memory simulation: a survey



Richard A. Uhlig, Trevor N. Mudge

June 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 2

Publisher: ACM Press

Full text available: pdf(636.11 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As the gap between processor and memory speeds continues to widen, methods for evaluating system designs before they are implemented in hardware are becoming increasingly important. method, trace-driven memory simulation, has been the subject of intense interest among researchers, as a result, enjoyed rapid development and substantial improvements during the past decade. This article surveys and analyzes these developments by establishing criteria for evaluating trace-driven simulation.

Keywords: TLBs, caches, memory management, memory simulation, trace-driven simulation

99 Report from the NSF workshop on workflow and process automation in information systems



Amit Sheth, Dimitrios Georgakopoulos, Stef M. M. Joosten, Marek Rusinkiewicz, Walt Scacchi, Jack Alexander L. Wolf

January 1997 **ACM SIGSOFT Software Engineering Notes**, Volume 22 Issue 1

Publisher: ACM Press

Full text available: pdf(1.24 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

An interdisciplinary research community needs to address challenging issues raised by applying workflow management technology in information systems. This conclusion results from the NSF workshop on Workflow and Process Automation in Information Systems which was held at the State Botanical Garden of Georgia during May 8-10, 1996. The workshop brought together active researchers and practitioners from several communities, with significant representation from database and distributed systems, software engineering, and information systems.

100 Task dependence and termination in Ada



Laura K. Dillon

January 1997 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 6 Issue 1

Publisher: ACM Press

Full text available: pdf(685.94 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [reviews](#)

This article analyzes the semantics of task dependence and termination in Ada. We use a control flow graph to model Ada tasking in examining the implications of and possible motivation for the rules that determine when procedures and tasks terminate during execution of an Ada program. The termination rules prevent data that belong to run-time instances of scope units from being deallocated prematurely, but they are unnecessarily conservative in this regard. For task instances that are created by i ...

Keywords: Ada tasking, distributed termination, master/dependent relation, task termination, execution model

Results 81 - 100 of 200


Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) **[5](#)** [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.


[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

72 Databases in software engineering: a roadmap

 Klaus R. Dittrich, Dimitrios Tombros, Andreas Geppert
May 2000 **Proceedings of the Conference on The Future of Software Engineering**


Publisher: ACM Press

Full text available:  [pdf\(1.33 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

73 Software maintenance and evolution: a roadmap

 Keith H. Bennett, Václav T. Rajlich
May 2000 **Proceedings of the Conference on The Future of Software Engineering**

Publisher: ACM Press

Full text available:  [pdf\(1.40 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

74 Formal specification: a roadmap

 Axel van Lamsweerde
May 2000 **Proceedings of the Conference on The Future of Software Engineering**

Publisher: ACM Press

Full text available:  [pdf\(1.30 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

75 Task force report and recommendations

 January 2000 **ACM SIGMIS Database**, Volume 31 Issue 1

Publisher: ACM Press



Full text available:  [pdf\(4.42 MB\)](#) Additional Information: [full citation](#), [index terms](#)

76 The use of multithreading for exception handling

Craig B. Zilles, Joel S. Emer, Gurindar S. Sohi


November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  [pdf\(1.49 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

Common hardware exceptions, when implemented by trapping, unnecessarily serialize program in dynamically scheduled superscalar processors. To avoid the consequences of trapping the main thread, multithreaded CPUs can exploit control and data independence by executing the exception in a separate hardware context. The main thread doesn't squash instructions after the excepting instruction, conserving fetch bandwidth and allowing execution of instructions in parallel ...

77 PRIME—toward process-integrated modeling environments: 1

 Klaus Pohl, Klaus Weidenhaupt, Ralf Dömges, Peter Haumer, Matthias Jarke, Ralf Klamma
October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume

Publisher: ACM Press

Full text available:  [pdf\(1.15 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Research in process-centered environments (PCEs) has focused on project management support neglected method guidance for the engineers performing the (software) engineering process. It dominated by the search for suitable process-modeling languages and enactment mechanisms.

consequences of process orientation on the computer-based engineering environments, i.e., the tools used during process performance, have been studied much less. In this article, we prese

Keywords: PRIME, method guidance, process modeling, process-centered environments, proce integrated environments, process-sensitive tools, tool integration, tool modeling

78 Reducing maintenance costs through the application of modern software architecture princ



Christine Hulse, Scott Edgerton, Michael Ubnoske, Louis Vazquez

September 1999 **ACM SIGAda Ada Letters , Proceedings of the 1999 annual ACM SIGAda int conference on Ada SIGAda '99**, Volume XIX Issue 3

Publisher: ACM Press

Full text available: pdf(796.32 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index ter](#)

Large software programs are usually long lived and continually evolve. Substantial maintenance often extended by engineers trying to understand the software prior to making changes. To suc evolve the software, a thorough understanding of the architect's intentions about software orga required. Software maintenance costs can be reduced significantly if the software architecture is defined, clearly documented, and creates an environment that promotes design consi ...

Keywords: architecture, design patterns, modeling, real-time software, software maintenance

79 Architectural framework modeling in telecommunication domain

Giulio Fregonese, Alessandro Zorer, Giovanni Cortese

May 1999 **Proceedings of the 21st international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.12 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: architectural patterns, design patterns, distributed systems, domain analysis, netw service management, network traffic data analysis, object-oriented framework, software archite software reuse

80 GENOA—a customizable, front-end-retargetable source code analysis framework



Premkumar T. Devanbu

April 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume :



Publisher: ACM Press

Full text available: pdf(241.27 KB)



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index ter](#)

Code analysis tools provide support for such software engineering tasks as program understand software metrics, testing, and reengineering. In this article we describe GENOA, the framework application generators such as Aria and GEN++ which have been used to generate a wide range practical code analysis tools. This experience illustrates front-end retargetability of GENOA; we the features of the GENOA framework that allow it to be ...


Keywords: code inspection, metrics, reverse engineering, source analysis

-  Tim Leehane
September 1996 **Proceedings of the 24th annual ACM SIGUCCS conference on User services**
Publisher: ACM Press
Full text available:  [pdf\(5.73 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

105 Reusable software components


-  Trudy Levine
July 1996 **ACM SIGAda Ada Letters**, Volume XVI Issue 4
Publisher: ACM Press
Full text available:  [pdf\(2.45 MB\)](#) Additional Information: [full citation](#), [index terms](#)

106 TIPSTER architecture: TIPSTER text phase II architecture concept

- Architecture Committee
May 1996 **Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996**
Publisher: Association for Computational Linguistics
Full text available:  [pdf\(1.28 MB\)](#) Additional Information: [full citation](#), [abstract](#)



The TIPSTER Architecture is a software architecture for providing Document Detection (i.e. Info Retrieval and Message Routing) and Information Extraction functions to text handling applications. The high level architecture is described in an Architecture Design Document. In May 1996, when the architecture design is complete, an Interface Control Document will be provided specifying the content of all inputs and outputs to the TIPSTER modules.

107 TIPSTER architecture: TIPSTER text phase II configuration management plan


- Architecture Committee
May 1996 **Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996**
Publisher: Association for Computational Linguistics
Full text available:  [pdf\(909.69 KB\)](#) Additional Information: [full citation](#), [abstract](#)

This document presents the TIPSTER Text Phase II Configuration Management (CM) Plan for identifying, controlling, and auditing the TIPSTER Architecture status and configuration definition.

108 Experiences with building distributed debuggers

-  Michael S. Meier, Kevan L. Miller, Donald P. Pazel, Josyula R. Rao, James R. Russell
January 1996 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**
Publisher: ACM Press
Full text available:  [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

109 DECALS: distributed experiment control and logging system

- Alex Hubbard, C. Murray Woodside, Cheryl Schramm
November 1995 **Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research**
Publisher: IBM Press
Full text available:  [pdf\(287.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



In developing distributed applications and services there is a need to be able to set up and run a set of processes. The experiment might be to obtain performance data, to test the processes' behavior or to evaluate an application management strategy. Common requirements are • to load and run versions of at least some of the software, often on multiple nodes of a network, • to initialize them in a well-controlled way, so the tests may be repeatable, • ...

110 A Model and a System for Data-Parallel Program Visualization

Thomas A. Wagner, R. Daniel Bergeron

October 1995 **Proceedings of the 6th conference on Visualization '95**

Publisher: IEEE Computer Society

Full text available:  [pdf\(1.50 MB\)](#) 
[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#)

Parallel program visualization and debugging require new techniques for gathering and displaying execution trace and profile data. Interaction with the program during execution is also required for parallel debugging. We discuss the difficulties associated with runtime user/program interaction in the data-parallel programming paradigm facilitates much more liberal runtime interaction than MIMD-based models. We present a model for data-parallel program visualization that ad ...

Keywords: visualization, parallel languages, SPMD, MIMD, SIMD

111 A compiler-directed distributed shared memory system



Tzi-cker Chiueh, Manish Verma

July 1995 **Proceedings of the 9th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.22 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)


112 Pen computing: a technology overview and a vision



André Meyer

July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(5.14 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This work gives an overview of a new technology that is attracting growing interest in public as the computer industry itself. The visible difference from other technologies is in the use of a pen as the primary means of interaction between a user and a machine, picking up the familiar pen interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions. Starting with a short historic ...

113 Implications of hierarchical N-body methods for multiprocessor architectures



Jaswinder Pal Singh, John L. Hennessy, Anoop Gupta

May 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 2

Publisher: ACM Press



Full text available:  [pdf\(4.66 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

To design effective large-scale multiprocessors, designers need to understand the characteristic applications that will use the machines. Application characteristics of particular interest include the ratio of communication relative to computation, the structure of the communication, and the local cache memory requirements, as well as how these characteristics scale with larger problems and machine architectures. The important class of applications is based on hierarchical N-body methods, where ...



Keywords: N-body methods, communication abstractions, locality, message passing, parallel architecture, parallel computer architecture, scaling, shared address space, shared memory

114 Supporting dynamic data structures on distributed-memory machines



Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren

-  March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume :
Publisher: ACM Press
Full text available:  [pdf\(2.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
- Compiling for distributed-memory machines has been a very active research area in recent years. In this work, we have concentrated on programs that use arrays as their primary data structures. To date, little work has been done to address the problem of supporting programs that use pointer-based data structures. The techniques developed for supporting SPMD execution of array-based programs take into account the fact that arrays are statically defined and directly addressable. Recursive data structures are also supported.
- Keywords:** dynamic data structures



115 User interface software tools

-  Brad A. Myers
March 1995 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 2 Issue 1
Publisher: ACM Press
Full text available:  [pdf\(3.25 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
- Almost as long as there have been user interfaces, there have been special software systems to help design and implement the user interface software. Many of these tools have demonstrated productivity gains for programmers, and have become important commercial products. Others have proven less successful at supporting the kinds of user interfaces people want to build. This article discusses the different kinds of user interface software tools, and investigates why some have succeeded and others have failed.
- Keywords:** interface builders, toolkits, user interface development environments, user interface design



116 A concurrency analysis tool suite for Ada programs: rationale, design, and preliminary experience

-  Michal Young, Richard N. Taylor, David L. Levine, Kari A. Nies, Debra Brodbeck
January 1995 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 4 Issue 1
Publisher: ACM Press
Full text available:  [pdf\(2.93 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
- Cats (Concurrency Analysis Tool Suite) is designed to satisfy several criteria: it must analyze implementation-level Ada source code and check user-specified conditions associated with program execution; it must be modularized in a fashion that supports flexible composition with other tool components; including integration with a variety of testing and analysis techniques; and its performance and accuracy must be sufficient for analysis of real application programs. Meeting these objectives is a challenging task.
- Keywords:** Ada, concurrency, software development environments, static analysis, tool integration

117 The case of the killer robot (part 2)

-  Richard G. Epstein
December 1994 **ACM SIGCAS Computers and Society**, Volume 24 Issue 4
Publisher: ACM Press
Full text available:  [pdf\(2.22 MB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)

118 An object-oriented, distributed architecture for large-scale Ada systems

-  Phillipe Kruchten, Christopher J. Thompson
November 1994 **Proceedings of the conference on TRI-Ada '94**
Publisher: ACM Press
Full text available:  [pdf\(1.14 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an architectural model ideally suited for the description of large, distributed and control systems. This model is organized around multiple dimensions (or views) of software architecture and is used to describe the software architecture of a family of automated air traffic systems currently under development by Hughes Aircraft of Canada. Some of the features of the systems are described, and in particular the mechanism used for transparent acc ...

119 Software understanding through integrated structural and run-time analysis

Kenny Wong

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  [pdf\(179.25 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The problem of software evolution is particularly acute. It requires a comprehensive understanding of the whole software system, including its architectural and run-time aspects. Reverse engineering techniques have traditionally focused on static information, using compiler-based technologies for lexical, structural, and semantic analysis. There is little emphasis on runtime information. However, developing and performing optimizing, testing, and debugging tasks for evolution require run ...


120 An annotated bibliography of interactive program steering



Weiming Gu, Jeffrey Vetter, Karsten Schwan

September 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 9

Publisher: ACM Press

Full text available:  [pdf\(1.24 MB\)](#)

Additional Information: [full citation](#), [citations](#), [index terms](#)

Results 101 - 120 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) **[6](#)** [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)